Modern Computational Statistics Final: Forecasting atmospheric carbon dioxide levels with RStan

Skye Hersh *CS 146 / Minerva Schools at KGI*

Using weekly atmospheric CO_2 measurements from the Mauna Loa Observatory in Hawaii (public dataset available through the Scripps Institute of Oceanography), I predict atmospheric carbon dioxide levels through 2058, which is 100 years since the Scripps CO_2 program first began taking measurements at Mauna Loa. I also infer the year by which we can expect with high probability that atmospheric CO_2 will exceed 450 ppm — the threshold over which we critically reduce our chance to stabilize the average global temperature. I evaluate and compare a quadratic (likelihood) model and an exponential (likelihood) model for the task. I propose priors for their parameters, explaining my reasoning, and then use RStan, an imperative probabilistic programming language, to arrive at the parameters' posterior predicted distributions. Having arrived at appropriate parameters, I use RStan to generate future predicted values for CO_2 in ppm. Code available on GitHub.

Keywords: climate forecasting, Bayesian modeling, MCMC, R, RStan

The dataset

The data, provided by the Scripps CO_2 program, features atmospheric in-situ CO_2 values measured weekly since March 29th, 1958.

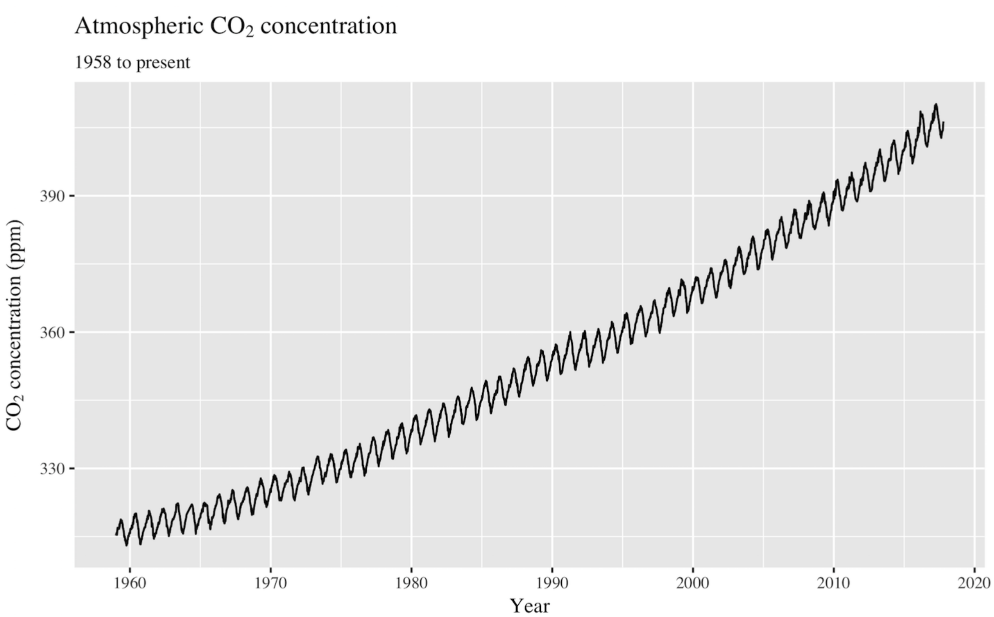

Figure 1: Original data

Data preprocessing

Though fairly consistent, I found that a few years were missing several values (though in 1964's case, 20 are missing). We'd want 51-53 values per year — usually 52, but sometimes 51 if the first measurement of the year occurs late in the week, sometimes 53 if the first year occurs early (as the exact number of days in a year is 365.25, we'd really expect ~52.18 weeks per year; this occasionally throws off the discrete count).

```
## [1] "Number of datapoints per year:"

##
## 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972
##   25   48   53   52   48   49   31   52   49   50   52   52   52   52   53
## 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987
##   52   52   52   51   53   52   52   52   52   52   53   48   51   52   52
## 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002
##   53   52   52   52   52   52   53   52   52   52   52   52   53   52   52
## 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
##   49   52   49   50   51   52   52   52   53   50   52   52   52   53   47
```

Where I could, I interpolated missing values by taking simple averages of the ppm values immediately neighboring them; considering the relative simplicity of the dataset, this strategy seems innocuous enough. In a few cases, there's a string of missing values: there, I make up the difference in equidistant intervals (there are only ever gaps on the side of a peak or trough, never where it seems a peak or trough is missing, so it seems fair to assume monotonically increasing or decreasing interpolated sequences in these cases).

Here are the graphs for 1961 and 1964, respectively: the former came in with a value for all 52 weeks, whereas the latter was missing 20 values (this is the most extreme case; most years with missing values were only missing 1 or 2).

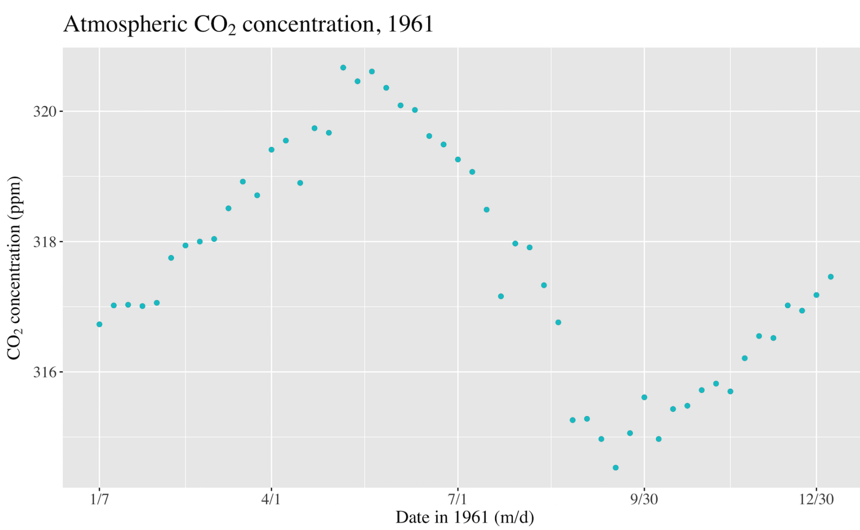

Figure 2: One year of data

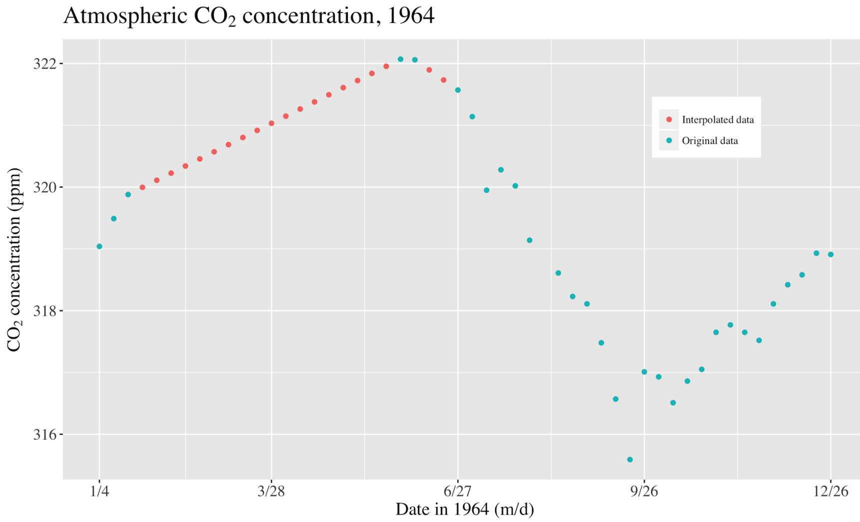

Figure 3: One year of data, with interpolations

I also generate new date values, at weekly intervals, to represent time steps up until 2058, for which we'll generate forecasted ppm values.

```
## [1] "Data points per year, after interpolating and generating future dates"

##
## 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972
##   25   52   53   52   52   51   51   52   52   52   52   52   52   52   53
## 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987
##   52   52   52   51   53   52   52   52   52   52   53   52   51   52   52
## 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002
##   53   52   52   52   52   52   53   52   52   52   52   52   53   52   52
## 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
##   52   52   53   52   51   52   52   52   53   52   52   52   52   53   52
## 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032
##   52   52   52   52   53   52   52   52   52   52   53   52   52   52   52
## 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047
##   53   52   52   52   52   52   53   52   52   52   52   53   52   52   52
## 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058
##   52   52   53   52   52   52   52   52   53   52    1
```

To permit models that evaluate ppm as a function of time step `t`, I assigned integer values to each date, representing days elapsed since the first date available in the data, `1958-03-29`: i.e., `1958-03-29` → `0`, `1958-04-05` (the next date, representing the CO_2 measurement taken one week after the first) → `7`, `1958-04-12` → `14`, etc.. I'll call these `day` values.

The range of ppm values is `313.04` to `410.18`; day values, `0` to `21791`. To simplify the assignment of priors for each parameter in the models, I normalized the data — both ppm and day values — to reside in the interval $[0, 1]$.

Finally, I separate the data (not including generated future timesteps) into a training and test set by an 80/20 split.

Model development and prior selection

I begin with the assumption that an appropriate model can be decomposed into a long-term trend to account for general trajectory, a seasonal trend to account for peaks during northern hemisphere winters and troughs during northern hemisphere summers (see *CO_2 seasonality pattern* below), and an noise component.

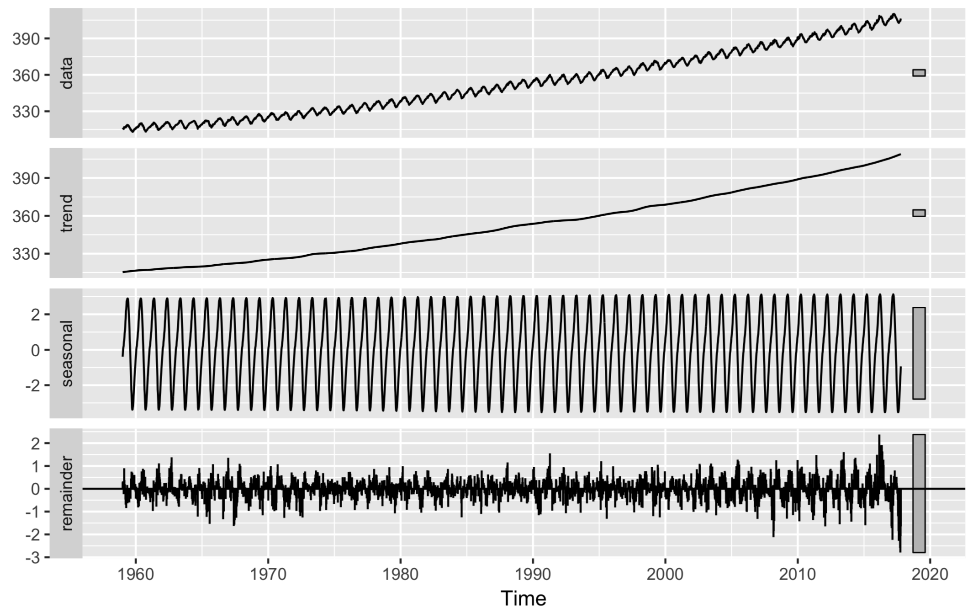

Figure 4: Decompositions

Long-term trend

By smoothing the time series by taking a moving average with length = seasonal span, we find something other than a linear long-term trend.

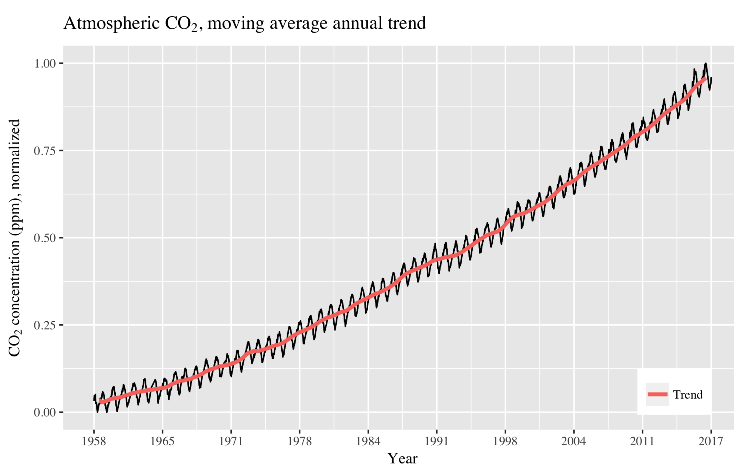

Figure 5: Moving averages

Just to check them out, I fitting second to fourth order polynomials with `R`'s linear modeling functionality. All shared an R^2 score exceeding 0.98.

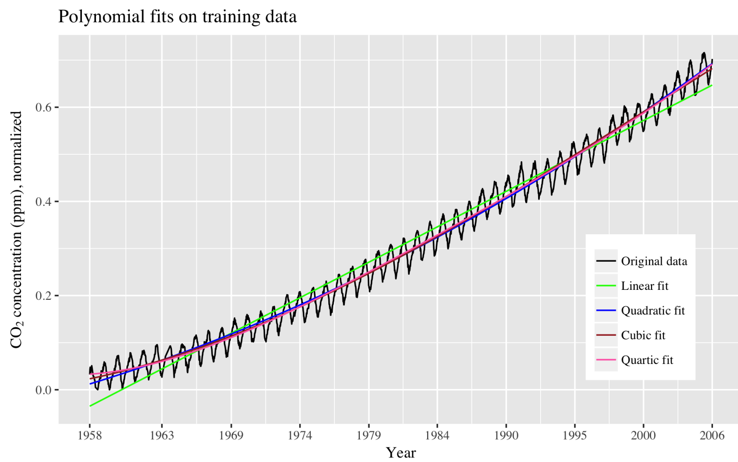

Figure 6: Polynomial fits

I tried several models for the long-term component, including 2nd order polynomial, 3rd order polynomial, and exponential functions. I found that the 3rd order polynomial (i.e., cubic) model failed to outperform the 2nd order (i.e., quadratic) one; I don't include analysis of a cubic model in this report.

Seasonal model

We see that CO_2 peaks in May of each year, and hits its low in September. Why does the peak occur so late in the spring? Presumably, the boreal and temperature forests of the world's largest forested landmass — Siberia — don't swing back into full photosynthetic production until then.

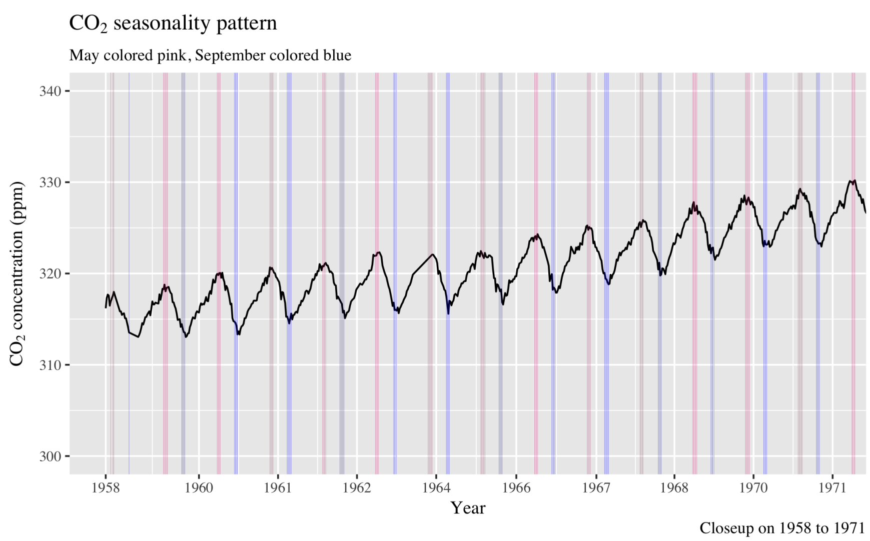

Figure 7: Seasonal peaks and troughs

If we subtract the long-term trend from the data, we find a very consistent seasonal trend (see next page). The following graph shows the data plotted against the seasons in each year. We see what looks like slowly increasing distances between each year over time, and affirm the uniformity of peaks in May (around week

21) and troughs in September (around week 40).

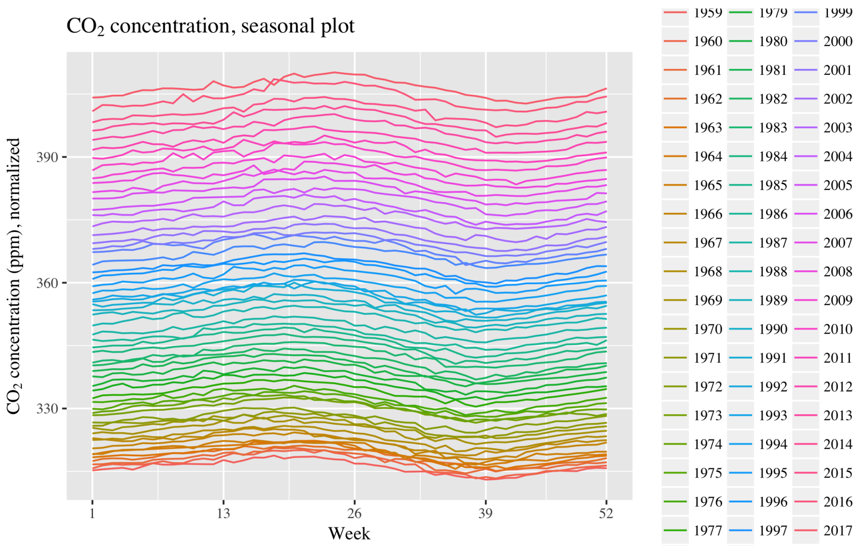

Figure 8: Seasonal trend, by year

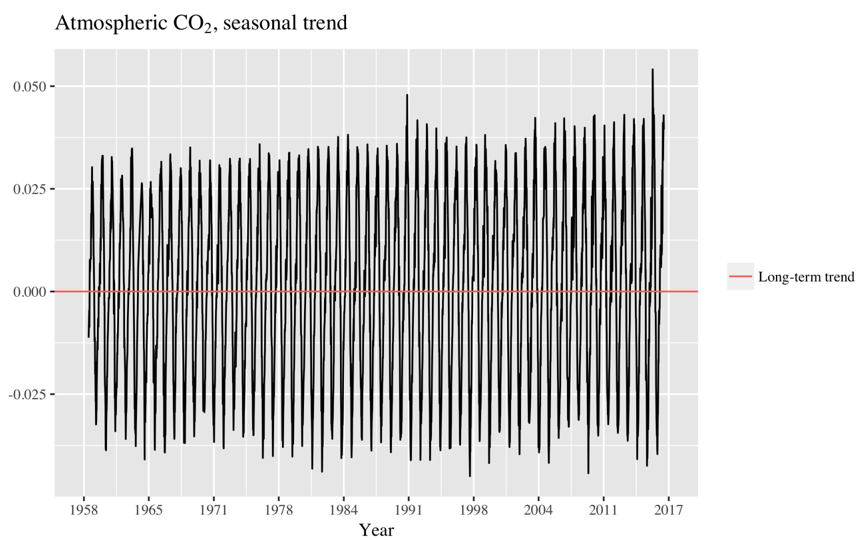

Figure 9: Seasonal trend, over years

I elect to move forward with a trigonometric periodic function to model seasonality (I'll use a cosine function), but this assumption may be limiting: the seasonal trend above seems to indicate that seasonal peaks are actually getting more extreme over time (interestingly, troughs seem consistent), and the static amplitude of a cosine function will fail to capture that.

Noise model

I make the assumption of normally distributed noise. This is an easy assumption to make, and further work might try a Gamma distribution.

Final models

I propose the following two models.

The **quadratic model** entails the likelihood function

$$p(ppm_i \mid c_0, c_1, c_2, A, \phi, \sigma) = N(c_0 + c_1 t_i + c_2 t_i^2 + A(cos(\frac{21791(2\pi t_i)}{365.25} + \phi), \sigma)$$

where CO_2 concentration at time t_i, ppm_i, is normally distributed around a quadratic trend parameterized by c_0, c_1, and c_2, a seasonal component parameterized by amplitude A and phase ϕ, and with a standard deviation of σ. Note that we divide the period of the seasonal trend by 365.25 — the number of days in the year — to describe the number of days in each period, and that we multiply is by 21,791 — the number of days, total, that comprise the dataset — to account for the normalization on the interval $[0, 1]$.

The **exponential model** entails the likelihood function

$$p(ppm_i \mid c_0, c_1, c_2, A, \phi, \sigma) = N(c_0 + c_1 c_2^{t_i} + A(cos(\frac{21791(2\pi t_i)}{365.25} + \phi), \sigma)$$

where, instead of a quadratic long-term trend, the term $c_1 c_2^{t_i}$ introduces an exponential function. Otherwise, the models are similar.

Prior selection

Quadratic model priors

In order to arrive at the quadratic model's priors, I considered the intervals in which I thought I could reasonably expect each parameter to live. With *ppm* values bound between 0 and 1, I'd start with hard bounds on σ, the error term, and A, the amplitude of the seasonal component. At the very least, all of these would have to be bound on $[0, 1]$ themselves (I arrive at more specific priors shortly).

I assign the phase parameter, ϕ, a Normal prior $N(0, \frac{\pi}{2})$, suggesting a lower bound of $-\pi$ and upper bound of π (I tried a uniform prior over this interval initially, but ran into sampling errors; the Normal prior worked decently). As for c_1 and c_2, the trend coefficients, I have no *a priori* reason to assume positive values (though they obviously are). In the interest of rigor, I don't require that they be positive. The intercept term, c_0, should be somewhere around 0. I give them all Normal priors, and for a sanity check, I sampled from the Normals that I hypothesized might define them:

$$p(c_0) \sim N(0.02, 0.01)$$

$$p(c_1) \sim N(0.45, 0.10)$$

$$p(c_2) \sim N(0.50, 0.10)$$

They seem reasonable:

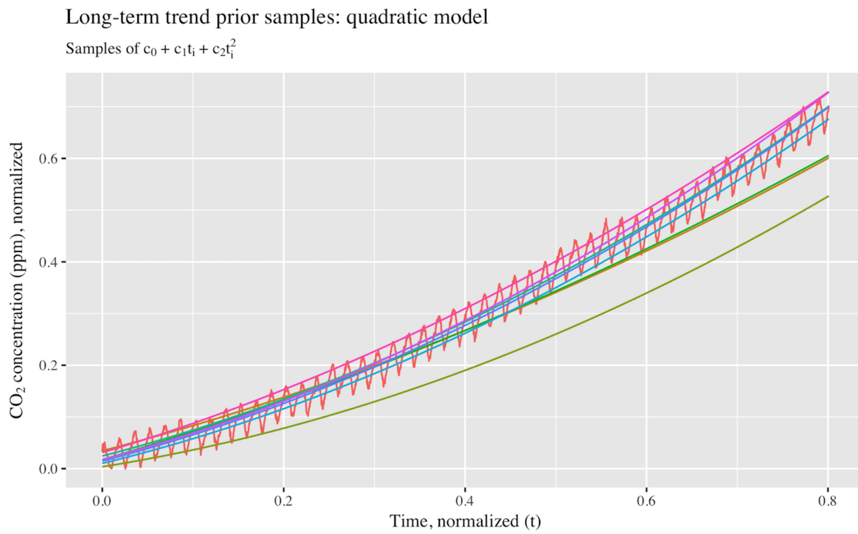

Figure 10: Quadratic model prior samples

I knew that A and σ, the amplitude and noise parameters, would necessarily be positive. Therefore, I transform these parameters in `Stan` exponentially to require them to be positive (I could explicitly require them to have a lower bound of 0, but to do this could interfere with the quality of Stan's sampling). I assumed that A might exist on $[0.01, 0.1]$, so I arrived at a suitable Normal prior like so:

$$\mu_A = \frac{\ln(0.01) + \ln(0.1)}{0.2} \approx -3.45$$

and

$$\sigma_A = \frac{\ln(0.1) - (-3.45)}{2} \approx 0.58$$

Thus, I arrive at a log transformed prior $p(A) \sim N(-3.45, 0.58)$. Assuming that σ might sit on $[0.001, 0.05]$, and following a similar procedure, I arrive at a prior

$$p(\sigma) \sim N(-4.95, 0.98).$$

Exponential model priors

I selected this latter models' prior transformations in more of an ad-hoc fashion: I tried a few things experimentally to see what would yield the best results. A, ϕ, and σ stay the same. However, I handle the trend coefficients — c_0, c_1, and c_2 — differently. Out of curiosity, I tranformed c_0 with a *tanh* function, to bound it to $[-1, 1]$, which I found worked adequately. Without requiring c_1 and c_2 to be positive, I found suboptimal results: I do, in this case, transform them with a sigmoid and exponential function, respectively.

Manually, I tried sampling these parameters from different Normal priors until I found a grouping that looked appropriate (perhaps not a very scientific method, but who's a purist here?):

$$p(c_0) \sim N(-0.7, 0.01)$$

$$p(c_1) \sim N(0.4, 0.05)$$

$$p(c_2) \sim N(0.95, 0.05)$$

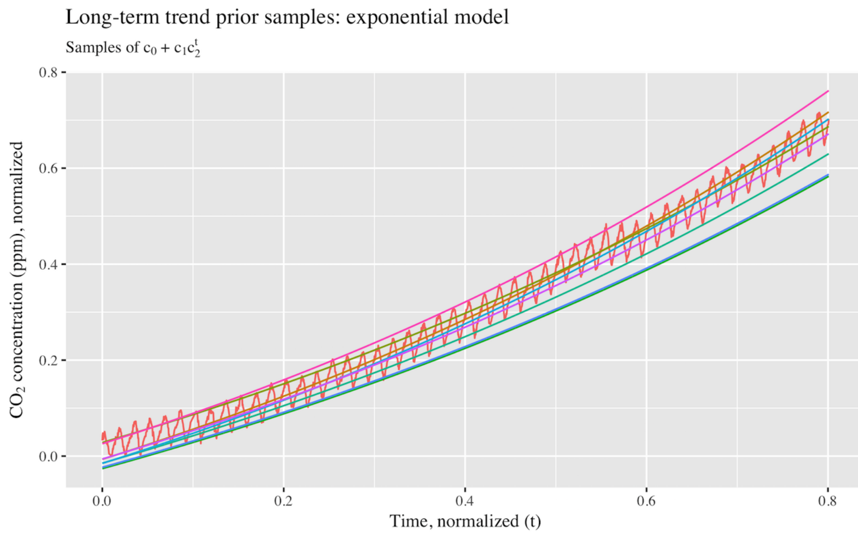

Figure 11: Exponential model prior samples

So ultimately, we arrive at the following two model proposals:

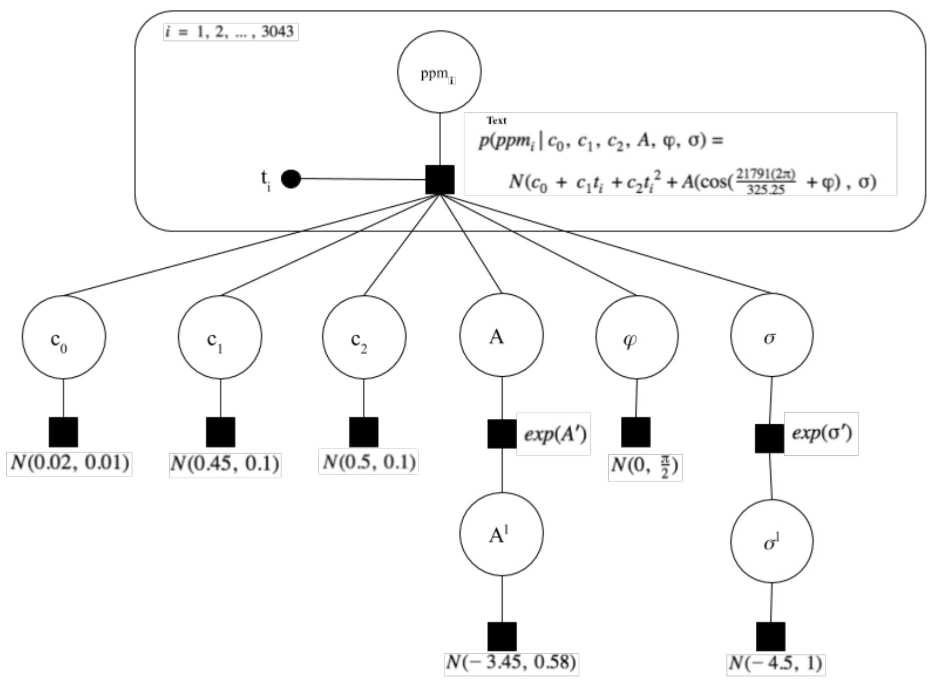

Figure 12: Quadratic likelihood graphical model.

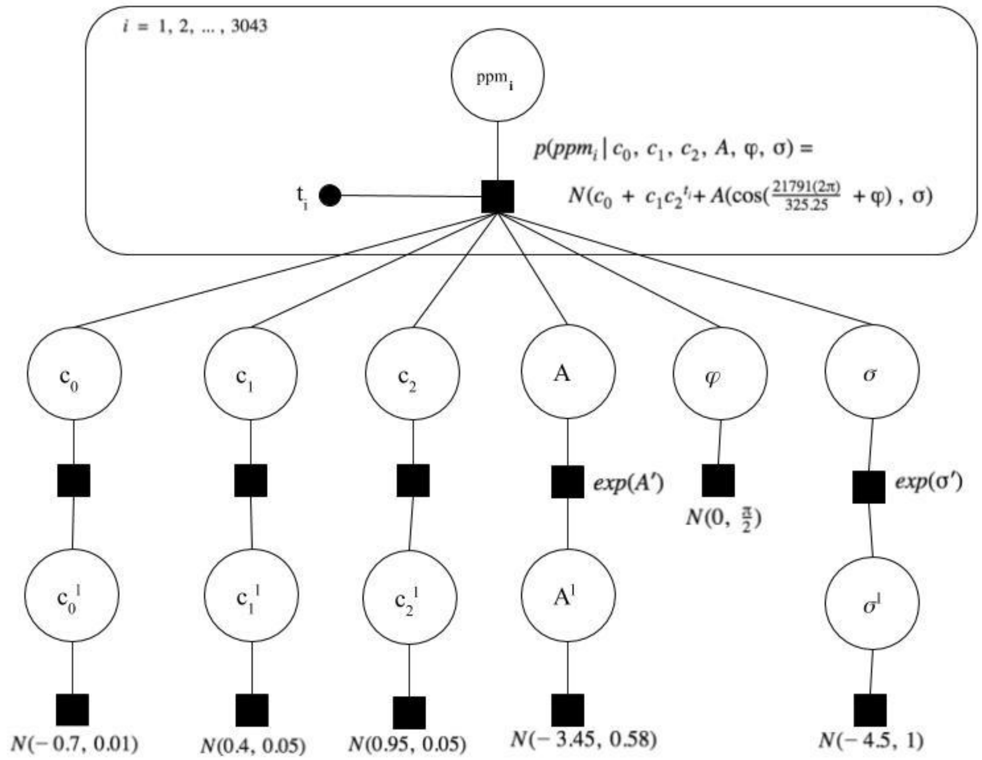

Figure 13: Exponential likelihood graphical model.

Modeling with Stan

I passed the likelihood function and priors to a Stan program, which uses Hamiltonian Monte Carlo sampling, a form of Markov chain Monte Carlo sampling, to verge on the underlying distributions for each parameter.

Both Stan programs converged nicely: all parameters have an `Rhat` of 1.0, which means that the ensemble of Markov chains for each program consistently converged (I used 3 chains per model).

Quadratic model sampling results

Parameter	mean	se_mean	sd	5%	95%	eff. sample size	Rhat
c0	0.01	0	0.00	0.01	0.01	2129	1
c1	0.51	0	0.00	0.51	0.51	1836	1
c2	0.43	0	0.00	0.42	2.44	1855	1
A	0.03	0	0.00	0.03	0.03	1907	1
phi	-0.43	0	0.00	-0.45	-0.42	2013	1
sigma	0.01	0	0.00	0.01	0.01	2079	1

Exponential model sampling results

Parameter	mean	se_mean	sd	5%	95%	eff. sample size	Rhat
c0	-0.59	0	0.01	-0.60	-0.58	983	1
c1	0.60	0	0.00	0.59	0.61	983	1

Parameter	mean	se_mean	sd	5%	95%	eff. sample size	Rhat
c2	2.60	0	0.01	2.58	2.63	993	1
A	0.03	0	0.00	0.03	0.03	2178	1
phi	-0.43	0	0.01	-0.45	-0.41	2160	1
sigma	0.01	0	0.00	0.01	0.01	2075	1

We can visualize the sampling quality with an autocorrelation function for each parameter; for brevity's sake, I only include here the ACFs for the quadratic model's parameter samples, but those for the exponential model look much the same. As evidenced below, the samples for each parameter are uncorrelated.

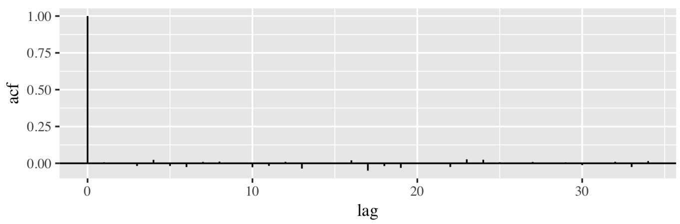

Figure 14: Autocorrelation of c0 samples

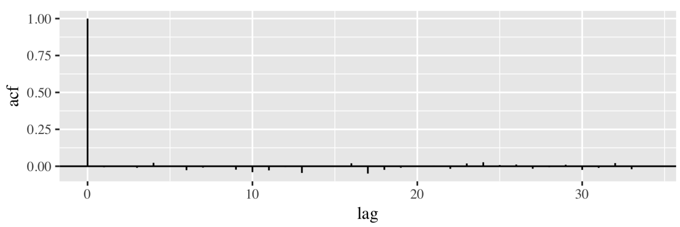

Figure 15: Autocorrelation of c1 samples

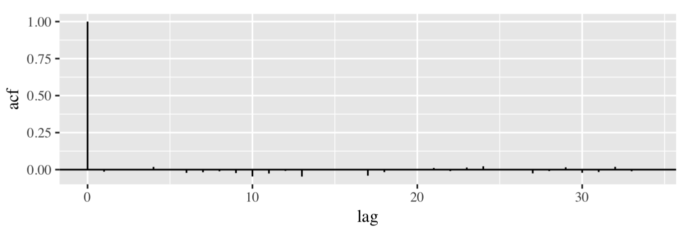

Figure 16: Autocorrelation of c2 samples

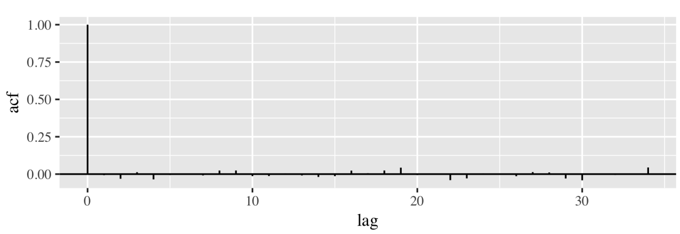

Figure 17: "Autocorrelation of A samples"

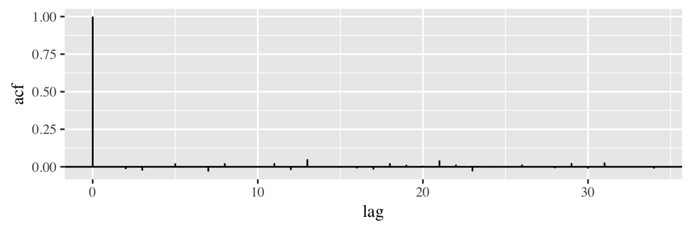

Figure 18: Autocorrelation of phi samples

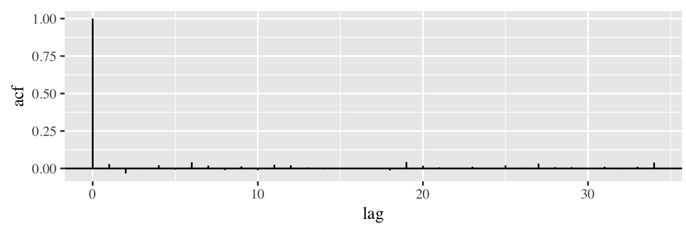

Figure 19: Autocorrelation of sigma samples

Finally, we see the 95% confidence intervals over the posterior predicted samples for the training dates (1958-2006) and test dates (2006-2017).

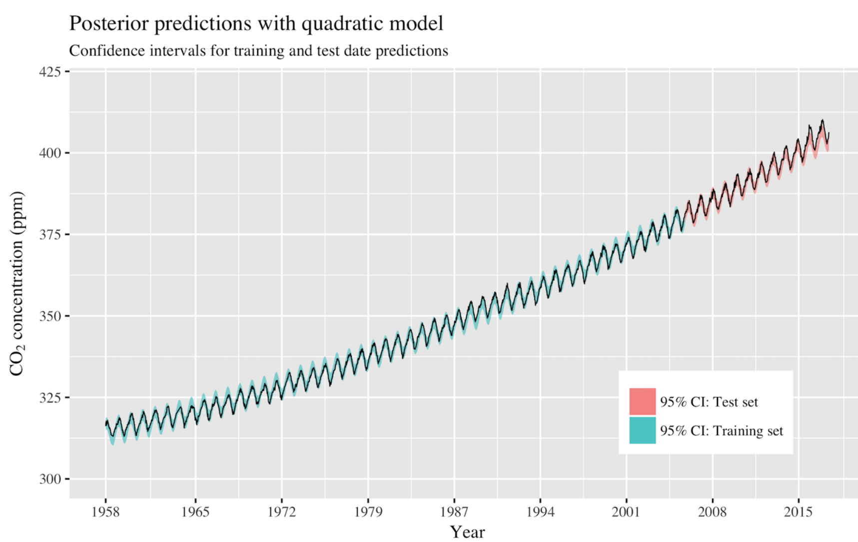

Figure 20: Quadratic model posterior sampling

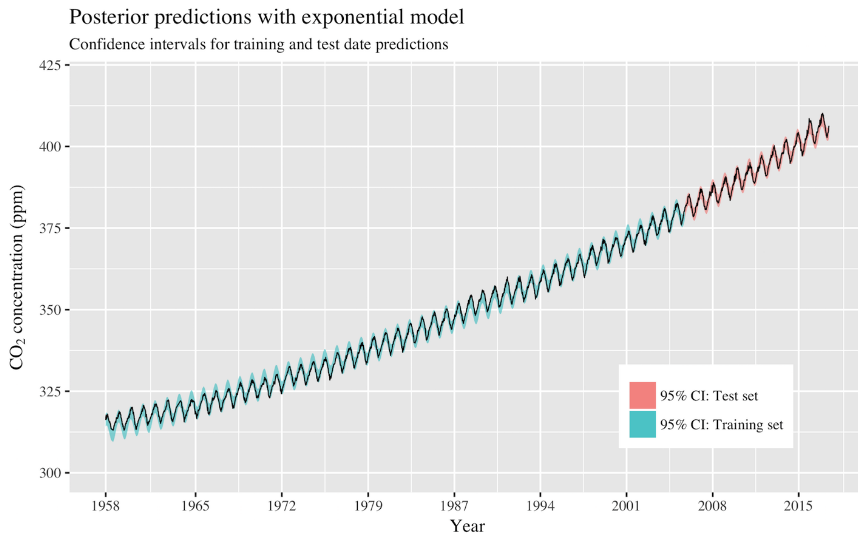

Figure 21: Exponential model posterior sampling

The quadratic model has a root mean square error (RMSE) of 1.32 on the test set, whereas the exponential model has only 0.91. But furthermore, I'm interested in the marginal likelihood of the data, given each model. I used the `bridgesampling` package to compute the (log) marginal likelihood of each model, re-fitting with `Stan` on the *full* dataset (as the marginal likelihood represents how well the fit explains *all*, as opposed to some of the data, I don't want to lose the information contained in the test set dates).

Analysis with `bridgesampling` suggests that the estimated Bayes factor — which quantifies the evidence for one hypothesis over another — in favor of the quadratic model over the exponential is equal to 3.094133e+22. I tend to prefer the exponential one, because its forecast suits my pessimism (it nearly hits 550 ppm by 2058):

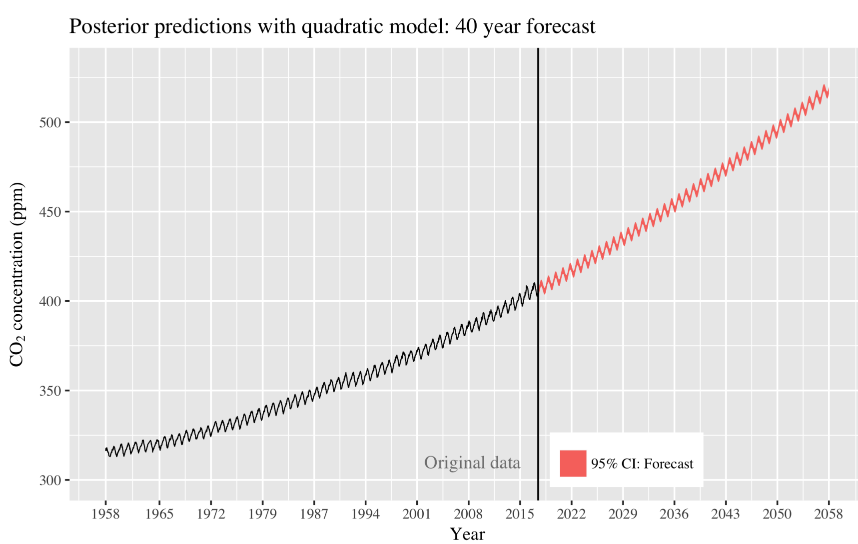

Figure 22: Quadratic forecast

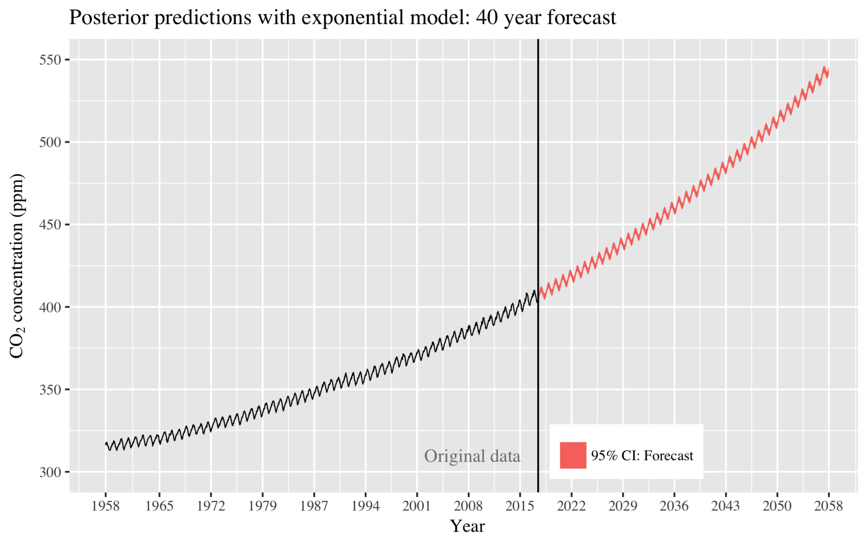

Figure 23: Exponential forecast

But considering the higher marginal likelihood with the quadratic model, I make the following predictions using that one.

Predictions

On January 5th, 2058, the 95% confidence interval for atmospheric CO_2 has a lower bound of **516.0349** ppm, and an upper bound of **519.9429** ppm. Furthermore, the 95% confidence interval bounds 450 ppm at the dates February 18th, 2034, and March 17th, 2035. Formally, this means that in 100 experiments with the same amount of samples, we could expect 95 to contain the true value in their confidence intervals. Colloquially, it means that the year of 2034 is looking pretty dangerous: once we hit 450 ppm, we critically diminish our chance at limiting global warming to 2 degrees Celsius.

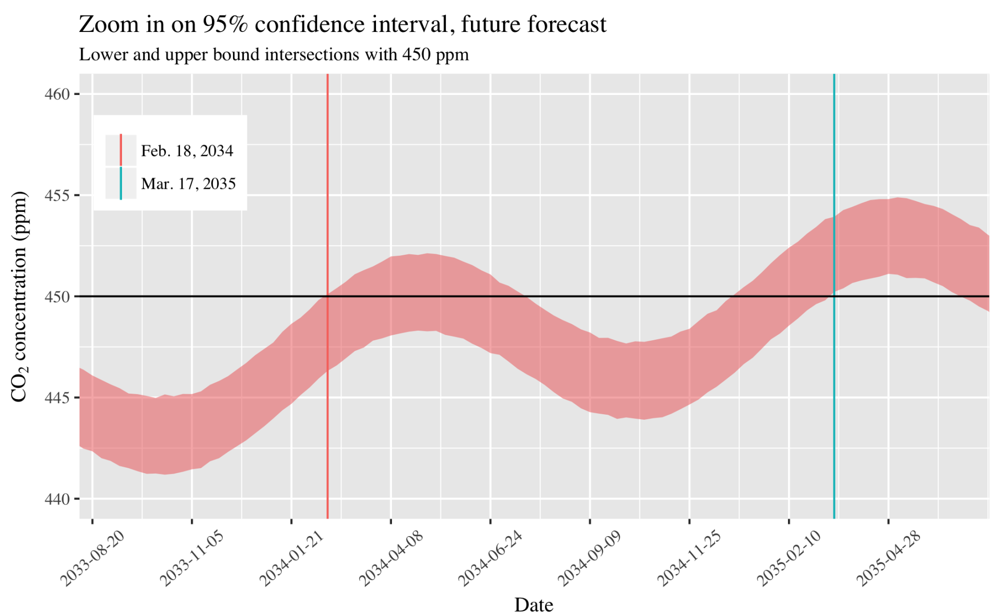

Figure 24: Upper and lower confidence bounds, 450 ppm